#2

Patent Office
Canberra

I, JULIE BILLINGSLEY, TEAM LEADER EXAMINATION SUPPORT AND SALES hereby certify that annexed is a true copy of the Provisional specification in connection with Application No. PS 2410 for a patent by TMG INTERNATIONAL HOLDINGS PTY LIMITED as filed on 20 May 2002.

WITNESS my hand this
Thirtieth day of May 2003

J. Billingsley

JULIE BILLINGSLEY
TEAM LEADER EXAMINATION
SUPPORT AND SALES

PRIORITY DOCUMENT
SUBMITTED OR TRANSMITTED IN
COMPLIANCE WITH RULE 17.1(a) OR (b)

AUSTRALIA

PATENTS ACT 1990

PROVISIONAL SPECIFICATION

FOR THE INVENTION ENTITLED:-

"SCHEDULING METHOD AND SYSTEM FOR RAIL NETWORKS"

The invention is described in the following statement:-

FIELD OF THE INVENTION

The present invention provides a method and associated system that can calculate a plan for moving trains on a network that reduces the cost of delays or late running. The invention is useful for timetable development, for real-time dynamic rescheduling of the trains on a network, and for assessing proposed changes to network infrastructure.

BACKGROUND TO THE INVENTION

Any discussion of the prior art throughout the specification should in no way be considered as an admission that such prior art is widely known or forms part of common general knowledge in the field.

In order for the scheduling tool to be useful as a realistic model of railway operations it must have the following features:

- The capability of representing a wide range of railway configurations including uni- and bi-directional track, junctions, branches, refuging or cross facilities etc.
- The ability to handle same direction overtakes.
- A check of the length of a train against the length of the crossing loop before allowing a cross to occur.
- The ability to take into account the characteristics of the signalling and safeworking systems.
- Accommodate all safety margins between the crossing and overtaking of trains to allow for signal changes.
- Allow trains to follow one another onto single line segments as governed by the signalling system in place.

It is an object of the present invention to overcome or ameliorate at least one of the disadvantages of the prior art, or to provide a useful alternative.

DETAILED DESCRIPTION OF THE INVENTION

The present invention seeks to provide a set of functional requirements for a package of integrated railway modelling tools, developed in collaboration with a reference group comprising key railway organisations.

One aspect of the invention provides algorithms for an improved single train simulator that will use energy-efficient and customised driving strategies to calculate

performance parameters that will form a basis for the other tools in the integrated package.

A further aspect of the invention provides a method and system for determining the efficient movement of trains on a network and in particular the development of an efficient strategy for controlling a flight of trains travelling in the same direction along a corridor.

A further aspect of the invention provides methods for infrastructure planning, timetable planning and dynamic rescheduling.

Modelling a Rail Network

Consider a railway system with the rail tracks modelled as a set of junctions and sections of rail joining them. Our basic network unit of track is called a segment, defined below, and illustrated in Figure 1.

Definition 1 (track segment). A track segment is a length of track which cannot be occupied by opposing trains. It is eliminated by either track junctions or signal points. A diamond crossing is a segment. Track segments have a length and a default direction. By assigning a default direction to segments, this model of a rail track system is a directed graph. In the present invention, an aggregated model of this directed graph is employed where these basic segments are partitioned into functional stations and links between them. The station definition includes physical stations but extends to any segment where a train may be scheduled to stop in our scheduling procedure. For example a crossing loop becomes a functional station for our purposes, because we may hold a train on the loop to enable another train to pass. Trains are moved strictly from station to station in our scheduling procedure.

Definition 2 (station) A station is a subset of track segments, at one geographical location, where trains may make scheduled stops and whence despatch decisions are made.

Stations are connected to each other by sequences of track segments with each feasible alternative sequence defining a path. We call a feasible path joining two stations a link.

Definition 3 (link) A link is a sequence of track segments joining two stations in the track graph. The first and last segment in a link between stations s_i and s_j must be a track segment in station s_i and s_j respectively.

Although the track graph is a directed graph, by virtue of the assigned default directions, we admit any (undirected) path as a link. As we shall see later, it is possible and sometimes desirable, to route a train on a link traversing some segments in reverse direction.

We now superimpose trains onto the track digraph to model an operating railway system as a train network.

Definition 4 (train network) A train network is a track digraph, a set of trains and a set of mappings relating to train dynamics through the network. A train is an object with attributes such as direction of travel, length, station of origin and destination station.

The state of the network is a representation of the location of each train in the system. The scheduling process is represented by the changing state of the system at a sequence of discrete timepoints or stages when decisions are made.

Trains may leave and enter the system during the scheduling planning horizon. We locate trains yet to be introduced into the system at a virtual source station and transfer trains exiting the network into a virtual sink station.

The normal scheduling task is to move each train through the network from its origin to destination. A train moves from station to station on a sequence of links joining intermediate stations on its selected path. Our goal here is to develop a scheduling procedure which optimises some objective measure of system performance. In the next section we describe a procedure aimed at minimising aggregate train lateness or a class of similar performance measures.

Train Movements

While a track segment can only accommodate one train at a time for trains travelling in opposite directions, if a track segment has internal signalling then it may be able to accommodate more than one train moving in the same direction. These following trains must be kept separated from each other by some minimum distance. This is achieved by having a following clearance time which governs the minimum separation between the front of following trains at entry and exit from a segment. This following clearance time will be a function of such things as train speed, the signalling system used, safety margins required and train length. An opposing clearance time is also needed to govern the minimum time separation between the front of one train

leaving a segment and a train travelling in the reverse direction seeking to gain access to the same segment.

- segment running times, headway, junction clearance
- dispatch rules

Deadlock

The issue of potential deadlock complicates the dispatch of trains particularly along single line corridors with passing loops. A feasible schedule requires that all trains can reach their destination without any train backing up.

Definition 5 (Deadlock) A rail network is defined to be deadlocked if at least one train cannot reach its destination without one or more trains backing up. [?] address this problem, referring to it as "line block". However their method assumes that any train in the system can refuge on any vacant loop. This condition is not necessarily true for the class of network problem we are considering here. For example in our problems we have long trains that are not able to refuge on all of the loops in the network. The following simple network has two long trains, one short train, and a short crossing loop.

Trains A and C will not fit on segment 4. This system will deadlock if train C is moved onto 6. Consequently the authors have investigated a more general theory of deadlock avoidance but it is computationally demanding. For the current method we have implemented only a limited deadlock avoidance protocol. The first advantage is that we retain a rich set of potential schedules only discarding them at deadlock or near certain deadlock states. The second advantage is that the avoidance procedure is computationally efficient and speed of schedule generation is a requirement of the system. The problem space search method used to construct the schedules randomly perturbates segment running times of individual trains to influence the order in which trains are dispatched. With this randomised search procedure it is good enough to reduce the occurrence of deadlock and simply throw away schedules that terminate in deadlock. The only small disadvantage is the loss of some computational efficiency because we pursue some potential schedules further than necessary.

Optimisation

One aim of the present invention is to construct a timetable that minimises the operating costs of the rail network. These costs could include costs associated with:

- delays at crossing loops

- lateness at key locations along a train's route

Mixed Integer Programming

Our modelling of the problem as a mathematical program was inspired by the observation that this scheduling problem could (almost) be formulated as a job-shop problem, with track segments corresponding to machines and train journeys corresponding to sequences of job operations. A key difference between our scheduling problem and the standard job-shop problem is that whereas jobs can be removed from one machine and put into a queue for the next machine, trains cannot be removed from the track after completing one track segment but before starting the next.

Our mixed integer programming formulations were based on a suggestion by Palitha Welgama (CSIRO). The track is assumed to be a sequence of track segments, numbered 1 ... n in the outbound direction. Stations between adjacent segments are assumed to have unlimited capacity.

T is the set of trains

i is an index for outbound trains

j is an index for inbound trains

P_{lk} is the time taken for train l to traverse segment k

t_{lk} is the time train l starts segment k

We define the function

$$d_{lmk} = \begin{cases} 1 & \text{if train } l \text{ crosses segment } k \text{ before train } m \\ 0 & \text{otherwise} \end{cases}$$

The following precedence constraints ensure that the trains remain on each section of track for their minimum traversal time and are then allowed to proceed onto the next segment of track on their journey:

$$\begin{aligned} t_{ik} - t_{ih} &\geq p_{ih} \\ t_{jh} - t_{jk} &\geq p_{jk} \end{aligned}$$

The following interference constraints ensure that no two trains occupy a single section of track at the same time:

$$\begin{aligned} t_{lk} - t_{mk} + M d_{lmk} &\geq p_{mk} \qquad \forall l, m \in T, l \neq m \\ t_{mk} - t_{lk} + M\,[1 - d_{lmk}] &\geq p_{lk} \qquad \forall l, m \in T, l \neq m \end{aligned}$$

M is a large number.

We minimise the sum of the arrival times of individual trains. Minimising the make span allows non-critical trains to run slowly.

This mixed integer formulation has been implemented in GAMS and can solve larger problems than our initial formulation, described below. It has been used to solve problems with 14 trains and 10 track segments. The main problem with this formulation is that it is for a single line track only, and assumes infinite capacity at stations. However, for such problems it generates optimal solutions that can be used to check the results of other timetabling methods.

The second mixed integer program models a single line railway corridor with stations connected by sections of track. Both the stations and the track sections are considered as machines and the trains as jobs. Each machine has a given maximum capacity. The stations are considered as a single unit with capacity equal to the sum of the number of crossing loops and mainline track segments at each station. It is assumed that each train can use any track segment in a station.

A_{ij} arrival time of train i at machine j.

D_{ij} departure time of train i from machine j.

P_{ij} minimum time train i takes on machine j.

C_j capacity of machine j.

δ_{ijt} indicator function which is equal to 1 if train i is on machine j at time t.

Sequencing equations

$$A_{ij} = D_{i,j-1} \quad \text{for all outbound trains } \forall j$$
$$A_{ij} = D_{i,j+1} \quad \text{for all inbound trains } \forall j$$

Location indicator

Indicator to determine to location of train i at time time t.

$$\delta_{ijt} = \begin{cases} 1 & A_{ij} \le t \le D_{ij}, \\ 0 & \text{otherwise.} \end{cases}$$

$$\delta^{A}_{ijt} = \begin{cases} 1 & t > A_{ij}, \\ 0 & \text{otherwise.} \end{cases}$$

The location indicator δ_{ijt} is produced by the following five equations:

$$\begin{aligned} t - A_{ij} &\leq M\delta^A_{ijt} \\ t - A_{ij} &\geq -M(1-\delta^A_{ijt}) \\ t - D_{ij} &\leq M\delta^B_{ijt} \\ t - D_{ij} &\geq -M(1-\delta^B_{ijt}) \\ \delta_{ijt} &= \delta^A_{ijt} - \delta^B_{ijt} \end{aligned}$$

Capacity constraints

$$\sum_i \delta_{ijt} \leq c_j \quad \forall j,t$$

Minimum processing time constraints

$$D_{ij} - A_{ij} \geq p_{ij} \quad \forall i$$

The objective used is to minimise the makespan of the scheduling tasks. This second formulation is a non-linear program with discontinuous constraints. While easy to develop the model could not be solved by GAMS even for the simplest of examples.

Langrangean relaxation

Lagragian relaxation has long been recognised as an effective solution method for constrained optimisation. Many computationally hard problems complicated by a set of difficult constraints can be decomposed into problems with a simpler structure. In our railway scheduling model the track capacity constraints are removed from the constraint set and placed in the objective function by the use of lagrange multipliers. These multipliers can be interpreted as the cost for using the track at a particular time. The higher the price on a track segment at a particular time the less likely it is to be used by trains at that time. This relaxed for of the scheduling problem allows us to reduce the problem to a series of shortest path problems for individual trains on the network. Trains are scheduled through the rail network one at a time through the matrix of prices (Lagrange multipliers) along their least cost path irrespective of other trains in the network. The solution of the relaxed problem with section capacity constraints removed may result in an infeasible schedule. A heuristic method must then be employed to remove the infeasible train movements and produce a feasible schedule.

The Lagrangian relaxation method is a common approach for solving large scale integer programs and is based on computing the solution to the dual problem. The optimal value for the dual problem provides a lower bound to the optimal value to the original problem. The upper bound together with the solution to the dual can be used to provide a good approximate solution to the original problem.

This method is successful for solving large problems provided the difference between the optimal values of the primal and dual is small and the method for solving the dual provides sufficient information for generating a nearly optimal feasible solution to the primal problem.

The main advantage of the lagrangean relaxation approach is that it produces both upper and lower bounds on the value of the objective function.

Using this Lagrangian relaxation method on several test problems it was found that the Lagrange multipliers oscillated and hence the primal and dual solutions did not converge or become sufficiently close at any iteration.

Problem Space Search

A heuristic is a technique which seeks good (near optimal) solutions to a problem at a reasonable computational cost without being able to guarantee optimality or state how close to optimally a feasible solution is. A heuristic can be thought of as a set of rules defining decision priorities. The dispatch procedure used is a greedy heuristic which builds a schedule by selecting which train to move next based only on local information.

Our solution method here uses the technique of problem space search. The problem space search technique is effective in a range of combinatorial optimisation problems including job shop scheduling and resource constrained project scheduling The Problem Space Search method takes a fast problem specific heuristic and embeds it within a local search procedure. The definition of a search neighborhood is based on a heuristic problem pair (h, p) where h is the fast heuristic and p represents the problem data upon which decisions are made. Since a heuristic h is a mapping from a problem to a solution the pair (h, p) is an encoding of a particular solution. By perturbing the problem data p the dispatch procedure will generate alternative solutions within a neighbourhood governed by the size of the perturbations made to the problem data. In the construction of solutions built by the base heuristic from perturbed data we must use

the original problem data. That is the perturbed data is used only to alter the decision process made by the dispatch procedure.

Below is described the dispatch procedure which moves a given set of trains from origin to destination minimising an objective function that is related to the delays experienced by the trains in the network.

1. Form a schedulable set of trains consisting of all trains not at their destination that have at least one unoccupied link.
2. From this schedulable set pick the train with the earliest start time from its current location. Assume this selected train is travelling from station S_i to station S_j.
3. From a contender set of trains consisting of all trains that have as their next move a dispatch from station S_i to S_j and vice-versa.
4. From this contender set select the train with the earliest arrival time at its successor station (either station S_i to S_j).
5. For the selected train invoke the deadlock avoidance procedure. If this procedure accepts the train then go on to step 6. If the train is rejected then remove it from the schedulable set. If the schedulable set is not empty then return to step 2 otherwise go to 7.
6. Schedule the selected train over its chosen link to its successor station.
7. Return to step 1 until all trains are at their destination or the schedulable set is empty.

In the search for the optimal schedule we use a two phase approach. We first implement a simple problem space search by making perturbations on the variables start time, st_i, and finish time, ft_i. A train's start time st_i is the time at which it is able to be scheduled from its current location in the schedule. Like the start time the finish time is a dynamic variable associated with each train. It is equal to the time that train i may arrive at its next station if dispatched next over its preferred available link. This value is a lower bound on a set of possible finish times for train i with the system in its current state.

In the second phase we use a population of the best solutions from the first phase to help direct the search towards refining low cost solutions found in the first phase.

The start time is used by the dispatch procedure to determine which two stations we schedule between. The finish time is then used to determine which train will be dispatched next between these chosen stations. In phase I of the solution process we

perturb these two variables enabling the dispatch procedure to generate many different sequences of dispatch decisions. The size of ϵ of the perturbations added to the start and finish times is governed by the parameter α and is given by

$$\epsilon = \alpha . U(0,1)$$

where $U(0,1)$ is a uniformly distributed random variable on the interval [0,1]. At each decision point it is possible for any train with a start time within the interval $[st_{min}, st_{min} + \alpha]$, where st_{min} is the minimum of the start times of all trains not a there destination, to be selected as the next dispatch. Thus by increasing the size of the parameter α we also increase the size of the set of possible next moves. By using a large value for α less importance is placed on the locally optimal decision and the size of the solution space the procedure is able to explore in increased. On the other hand using a too small value for α restricts the alternative solutions that the procedure is capable of generating. Small values for α are not likely to alter the decision making data sufficiently to generate a different solution sequence. Through experimentation we have found an appropriate value for α to be half the average inter station running time for all trains on the network.

In the second phase of the search process we take the best n schedules and examine them in sufficient detail to identify the good decisions in each. We have determined that a key aspect of the performance of good schedules is whether they incorporate the best passing rules at each station. We therefore bias the search towards pairwise passing strategies at each station which yield schedules with good global performance.

Long-haul examples

The problem space search dispatcher has been tested on two Australian railway networks. The objective function is that is used to evaluate each schedule is the sum of the lateness of each train. The lateness of each train is given by the function

$$Z^i(a_{id}) = \begin{cases} 0 & a_{id} \leq a_i^* \\ a_{id} - a_i^* & a_{id} > a_i^* \end{cases}$$

where a_{id} is the actual arrival time of train i at is destination while a_i* is the desired arrival time. The problem space search dispatcher was coded in Pascal and run on a Unix workstation.

The first of the test problems is the Australian North Coast Railway which runs from Maitland to Murwillumbah. This track consists of a single line corridor which 68 crossing loops and covers a distance of some 750km. Forty two trains are to be schedules over the corridor - twenty one north bound and twenty one south bound. Some trains traverse the entire length of the railway corridor while others use varying portions of it.

A histogram of the results from both phase I and phase II of the problem space search can be found in figure 3. In both phases 3000 feasible schedules were constructed and the histograms have been plotted using buckets of 1000 seconds. The lowest cost schedule found in phase I had a cost of 152000 seconds while the overall best solution was found in phase II with a cost of 145000 seconds. As can be seen in figure 3 of the phase II distribution of solution costs has been significantly skewed towards the low cost end when compared with the results from phase I. The best solution found is represented as a train graph in figure 5. Current best practice for this same actual scheduling task on the North Coast line is 205000 seconds. This is shown as the vertical line in figure 3. Our procedure is therefore generating a raft of better schedules, the best being approximately 30% lower than current practice. The program took 10 minutes to run.

The second test problem is the Sydney to Melbourne corridor running between Cambelltown Sydney and Spencer Street Melbourne. The track from Cambelltown to Junee is double line track with crossing loops to allow overtaking while the section from Junee to Spencer Street is single line track with crossing loops. The track covers a distance of 750km and has 59 crossing loops. Daily 118 trains are to be schedules through variations lengths of the railway corridor.

The results from both phases of the problem space search are presented in figure 4. Once again 3000 feasible schedules were constructed in both phases with the best schedule being found in phase II. The minimum cost schedule found in phase I was 53660 seconds while in phase II the best one found had a cost of 52945 seconds. Note the effect of the weights in skewing the histogram in phase II towards the low cost region. Figure 6 shows the train graph of the best found solution. Current best practice

for the Sydney to Melbourne scheduling task has a cost of 85000 seconds which is shown as the vertical line in figure 4. For this larger problem the running time was 29 minutes.

In both of the above examples solutions within 10% of the best one found were generated in under 2 minutes.

Suburban Networks

The problem of optimisation of suburban train schedules is significantly different from the long haul systems reported earlier. While there is a common theme of minimising resources employed to achieve a desired level of performance, suburban systems are driven by the need to provide regular, repeatable service to passengers.

In order to understand the basic problem and develop solution methods we first consider a very simple prototype of the suburban systems commonly operated in Australia - a hub and spokes system operating on an hourly timetable with each route having a given service time. The service time for each route is the time to travel the route and return to the hub including station stopping times and change-over times at the hub or termini.

It has been impressed on us that it is desirable for suburban rail systems to operate on a regular, cyclic timetable. Suppose we are given the required service level for each route in terms of headways e.g for each route there is a market determined headway of a train every 10 or 12 or 15 or 20 ... minutes down to one every 60 minutes. The aim is to develop a schedule which has trains departing at fixed times within the hour each hour of normal service . (We concentrate on the base or normal service and assume peak hour services are superimposed on this timetable.) Assume, in this first model, there are no capacity constraints at the hub, any train can service any route, the routes are non-intersecting and there are no side conditions on train operations other than route service times and route headways.

The aim is to model this simplified system and find an optimal solution.

The optimisation objective is to minimise resource use in the operating system. We model resource use as the sum of the trains servicing the system multiplied by the time each train is not operating, i.e. whenever a train is waiting at a terminus or hub other than for considerations of service time. It is measure of when any train and its crew

are in the system but not delivering a service to customers because of time-table constraints. The best solution minimizes this not-in-service penalty.

Since headways and route service times are given, the only controls (variables) in the optimisation are the relative departure times of the services to the various routes.

A solution is defined by the departure times of each service (in minutes past the hour).

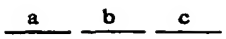
It was shown that this problem could be posed as a problem of cyclic groups.

Whilst the invention has been described with reference to a number of specific examples, it will be appreciated by those skilled in the art that the invention may be embodied in many other forms.

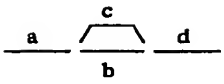
DATED this 20th day of May, 2002

TMG INTERNATIONAL HOLDINGS PTY LIMITED

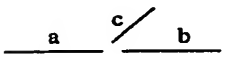
Attorney: RUSSELL J. DAVIES
Fellow Institute of Patent Attorneys of Australia
of BALDWIN SHELSTON WATERS



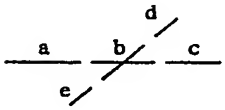
A station on a single line is modelled as a single track segment b. The set of possible movements on this network are {abc,cba}.



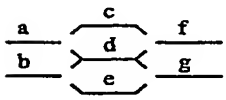
A loop on a single line is modelled as a pair of track segments b and c. The set of possible movements is {abd, acd, dba, dca}.



A junction can be modelled without additional tracks. The set of possible movements is {ab, ac, ba, ca}.



A diamond crossing requires an additional track segment, b, to be defined.



A centre refuge can be modelled using three track segments c, d, and e. If the refuge is on a double line then the set of movements is {acf, adf, gdb, geb}.

Figure 1: track segments

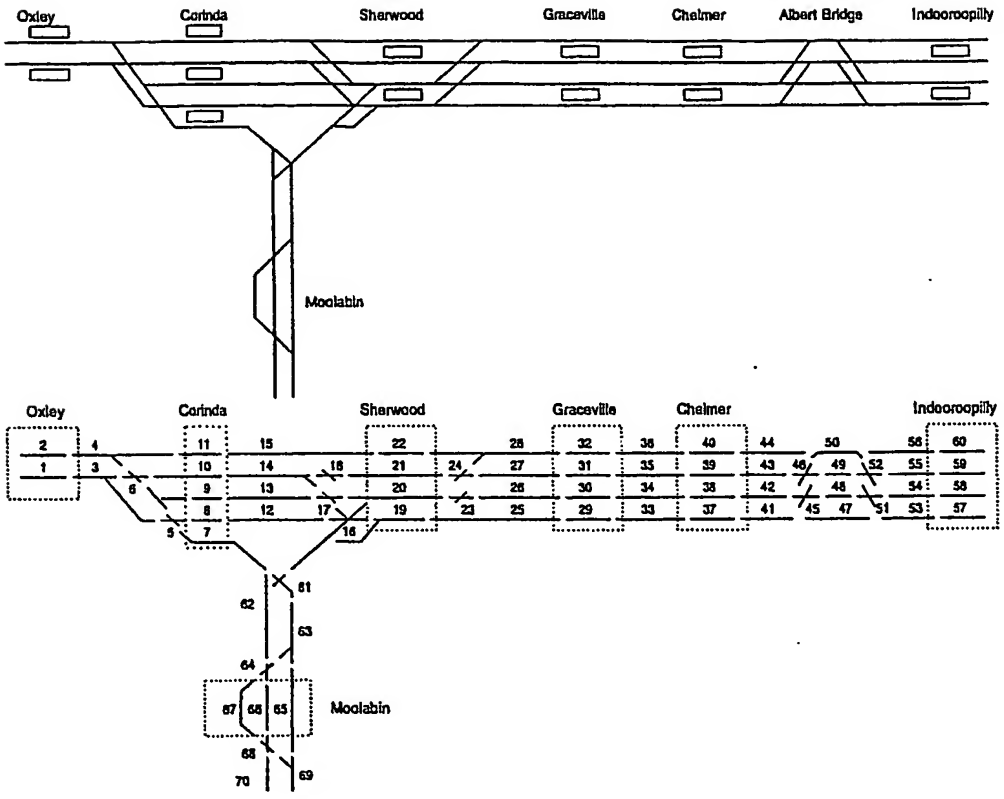


Figure 2: Network model showing segments and stations enclosed by broken lines

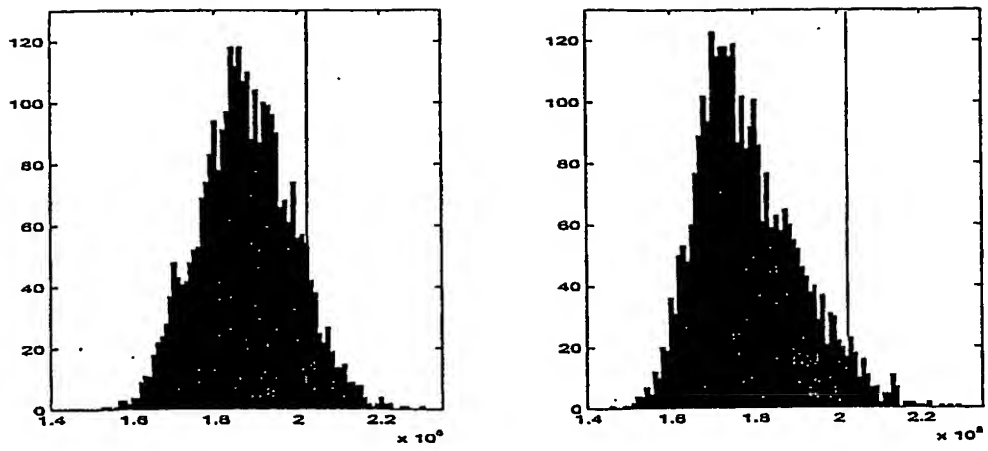


Figure 3: Results for the North Coast railway corridor

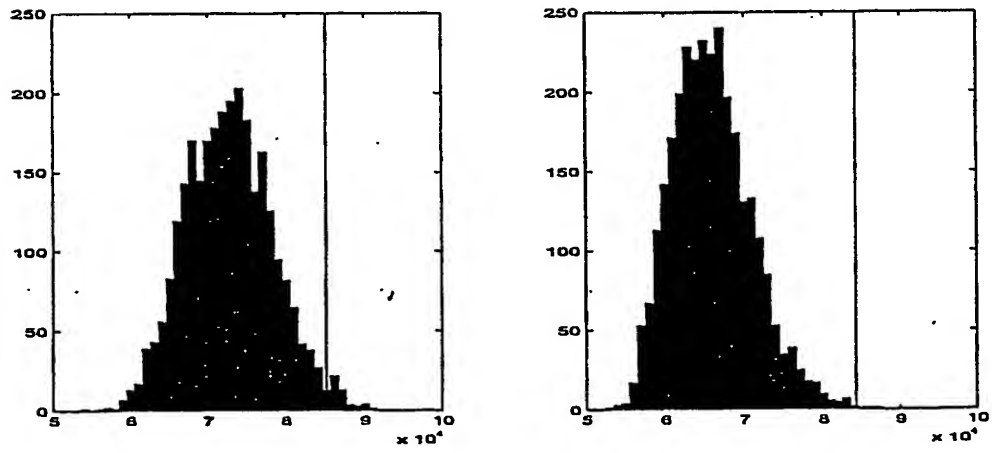


Figure 4: Results for the Sydney-Melbourne railway corridor

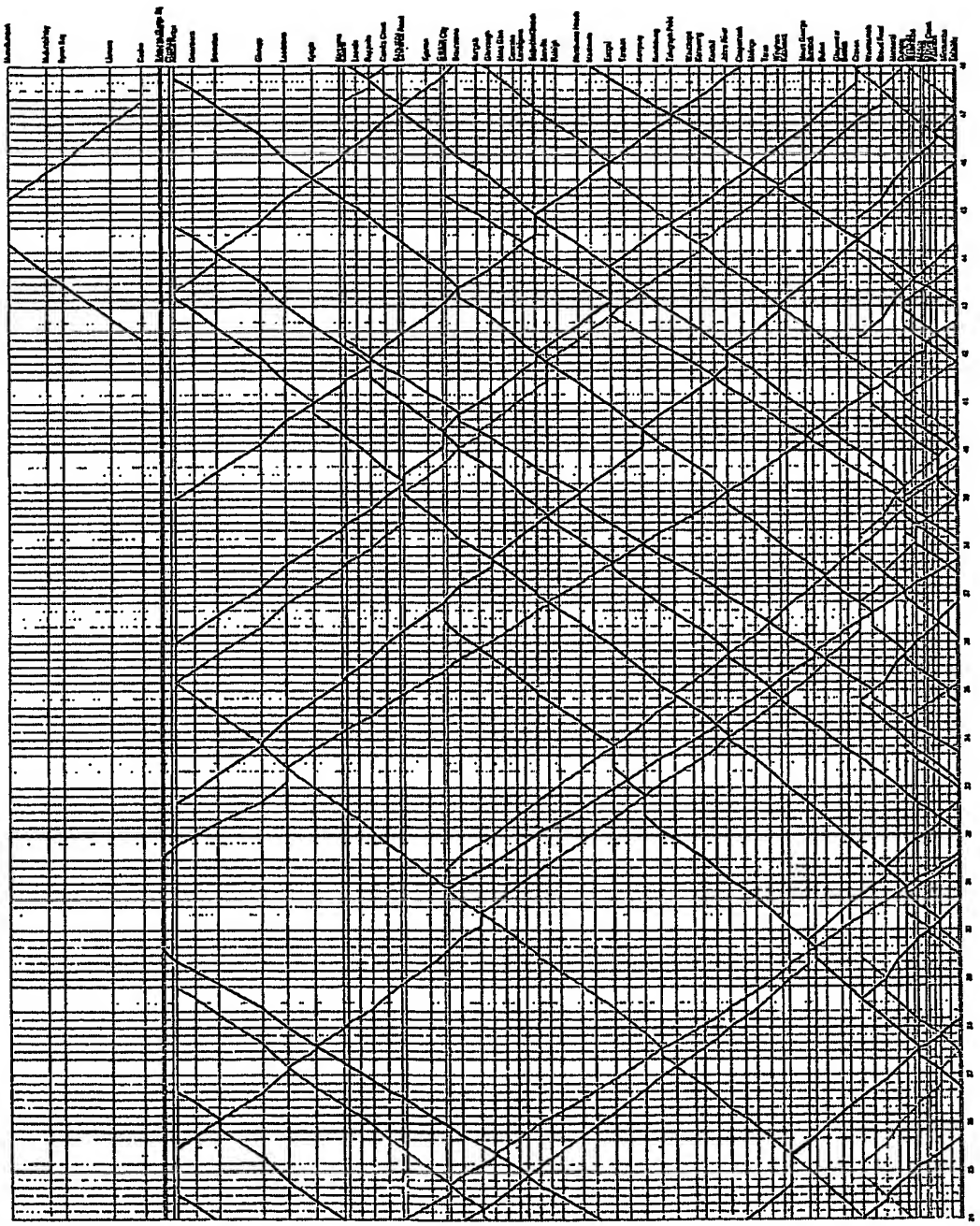


Figure 5: Crossing schedule for the North Coast railway corridor

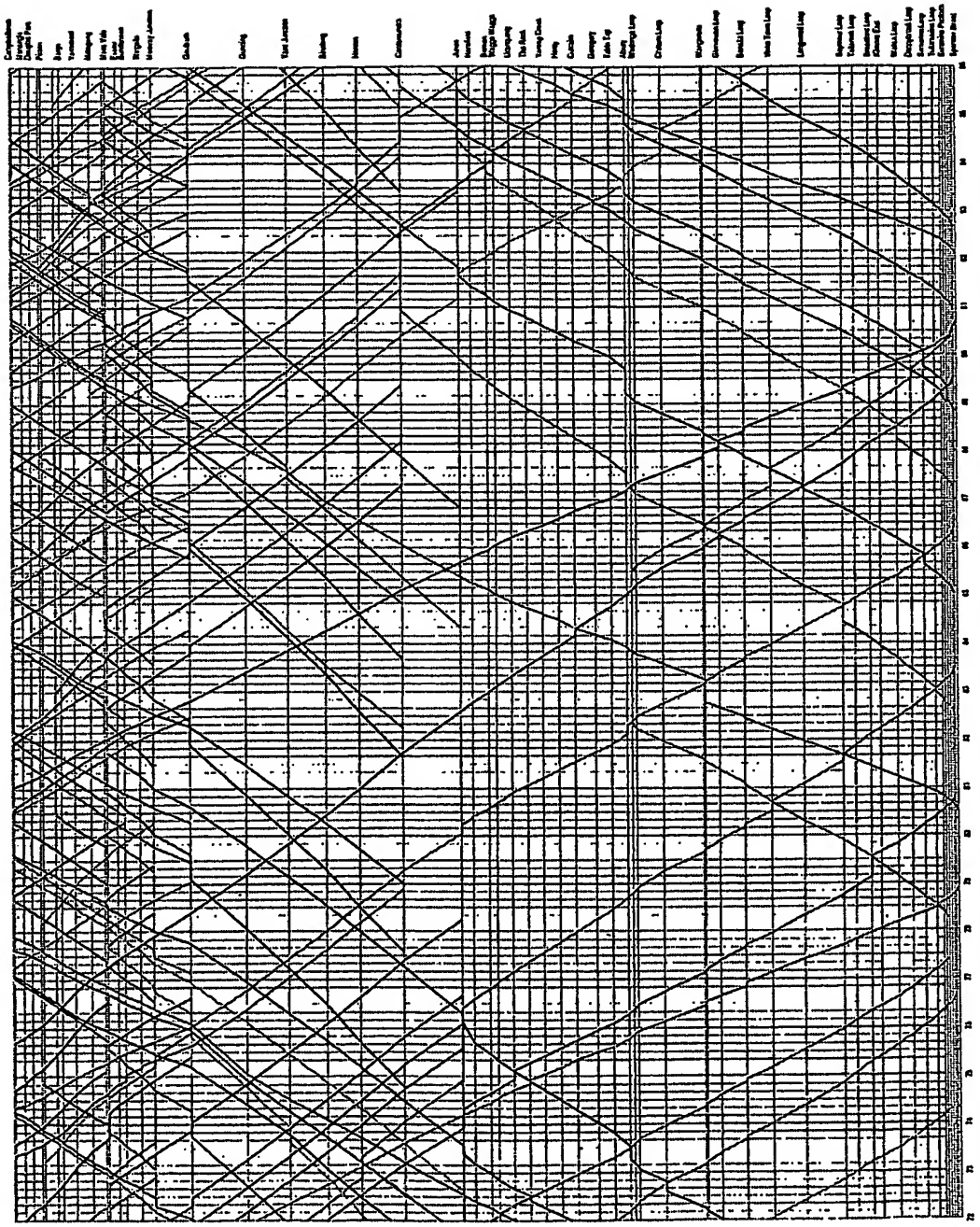


Figure 6: Crossing schedule for the Sydney - Melbourne railway corridor